

ANALISI API REST PROGETTO LABORATORIO

TABELLE

```
CREATE TABLE utenti(  
  idUtente int not null PRIMARY KEY auto_increment,  
  username varchar(30)not null unique,  
  nome varchar(30)not null,  
  cognome varchar(30)not null,  
  password varchar(32)not null,  
  codiceFiscale varchar(16)not null,  
  regione varchar(30)not null,  
  via varchar(30)not null,  
  nCivico varchar(5)not null )
```

```
CREATE TABLE richieste(  
  idRichiesta int not null PRIMARY KEY auto_increment,  
  rifUtente int not null,  
  oraInizioConsegna time not null,  
  oraFineConsegna time not null,  
  durataRichiesta time not null )
```

```
CREATE TABLE liste(  
  idLista int not null PRIMARY KEY auto_increment,  
  rifRichiesta int not null,  
  rifProdotto int not null,  
  Q.Tà int(10)not null )
```

```
ALTER TABLE richieste
```

ADD CONSTRAINT associazione1

FOREIGN KEY (rifUtente) REFERENCES utenti(idUtente)

ALTER TABLE liste

ADD CONSTRAINT associazione3

FOREIGN KEY (rifRichiesta) REFERENCES richieste(IdRichiesta)

ALTER TABLE liste

ADD CONSTRAINT associazione3

FOREIGN KEY (rifProdotto) REFERENCES prodotti(IdProdotto)

1)

GET /spesa/risposte				
Summary				
Elenco delle richieste completate				
Description				
Permette di avere l'elenco delle varie richieste soddisfatte/completate di tutti o di un determinato utente				
Parameters				
Name	Located In	Description	Required	Schema
rifUtente	Path	ID dell'utente del quale si vuole ottenere la lista di richieste completate	No	Xs:int
Responses				
Code	Description	Schema		
200	Elenco delle richieste completate dai vari utenti o da un unico utente	<xs:complexType name="return"> <xs:sequence> <xs:element name="risposte" minOccurs="0" maxOccurs="unbounded"> <xs:complexType> <xs:sequence> <xs:element name="rifUtente" type="xs:int"/> <xs:element name="idRisposta" type="xs:int"/> <xs:element name="rifRichiesta" type="xs:int"/> </xs:sequence> </xs:complexType> </xs:sequence> </xs:complexType>		

		</xs:element> </xs:sequence> </xs:complexType>
403	Non autorizzato	
406	Parametri di input non validi	
415	valore del content-type non è quello atteso	
Default	Errore non previsto	
Esempi output proposto all'utente dal consumer:		
//Select non utilizzata per proporre un output all'utente ma volta ad ottenere l'id identificativo della/delle richieste completate dall'utente		
Esempi query SQL:		
SELECT idRisposta, rifRichiesta WHERE rifUtente=[rifUtente]		
Esempi create table SQL:		
CREATE TABLE risposte (idRisposta int not null PRIMARY KEY auto_increment, rifUtente int not null, rifRichiesta int not null)		
ALTER TABLE risposte ADD CONSTRAINT associazione2 FOREIGN KEY (rifUtente) REFERENCES utenti(idUtente)		
ALTER TABLE risposte ADD CONSTRAINT associazione3 FOREIGN KEY (rifRichiesta) REFERENCES richieste(IdRichiesta)		

2)

GET /spesa/prodotto
Summary

Elenco dei prodotti				
Description				
Permette di ottenere un elenco di prodotti in base a determinati parametri				
Parameters				
Name	Located In	Description	Required	Schema
Genere	Query	Genere del prodotto	No	Xs:string
Etichetta	Query	Etichetta del prodotto	No	Xs:string
Costo	Query	Costo del prodotto	No	Xs:double
Nome	Query	Nome del prodotto	No	Xs:string
Marca	Query	Marca del prodotto	No	Xs:string
Descrizione	Query	Descrizione del prodotto	No	Xs:string
Responses				
Code	Description	Schema		
200	Elenco dei prodotti	<pre> <xs:complexType name="return"> <xs:sequence> <xs:element name="prodotto" minOccurs="0" maxOccurs="unbounded"> <xs:complexType> <xs:sequence> <xs:element name="idProdotto" type="xs:int"/> <xs:element name="Genere" type="xs:string"/> <xs:element name="Etichetta" type="xs:string"/> <xs:element name="Costo" type="xs:double"/> <xs:element name="Nome" type="xs:string"/> <xs:element name="Marca" type="xs:string"/> <xs:element name="Descrizione" type="xs:string"/> </xs:sequence> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType> </pre>		
403	Non autorizzato			
406	Parametri di input non validi			
415	valore del content-type non è quello atteso			
Default	Errore non previsto			

Esempi output proposto all'utente dal consumer:

Genere	Etichetta	Costo	Nome	Marca	Descrizione
Alimentare	Aaaaa	2,50	Pizza margherita	Bbbb	Una pizza margherita senza glutine

Esempi query SQL:

```
SELECT idProdotto, Genere, Etichetta, Costo, Nome, Marca, Descrizione  
FROM prodotti
```

Esempi create table SQL:

```
CREATE TABLE prodotti(  
    idProdotto int not null PRIMARY KEY auto_increment,  
    genere varchar(30) not null,  
    etichetta longtext not null,  
    costo double(6,2) not null,  
    nome varchar(50) not null,  
    marca varchar(50) not null,  
    descrizione longtext not null  
)
```

3)

POST /spesa/prodotto

Summary

Inserimento prodotto

Description

Permette di aggiungere un prodotto al database

Parameters

Name	Located In	Description	Required	Schema
Genere	Body	Genere del prodotto	Yes	Xs:string
Etichetta	Body	Etichetta del prodotto	Yes	Xs:string
Costo	Body	Costo del prodotto	Yes	Xs:double
Nome	Body	Nome del prodotto	Yes	Xs:string
Marca	Body	Marca del prodotto	Yes	Xs:string
Descrizione	Body	Descrizione del prodotto	Yes	Xs:string

Responses

Code	Description	Schema
200	Operazione eseguita correttamente	<xs:element name="return" type="xs:boolean">

403	Non autorizzato	
406	Parametri di input non validi	
415	valore del content-type non è quello atteso	
Default	Errore non previsto	
Esempi output proposto all'utente dal consumer:		
<ul style="list-style-type: none"> ➤ "Operazione andata a buon fine" ➤ "Errore, operazione non completata" 		
Esempi query SQL:		
INSERT INTO prodotto (genere, etichetta, costo, nome, marca, descrizione) VALUES (\$_POST[genereProdotto], \$_POST[etichettaProdotto], \$_POST[costoProdotto], \$_POST[nomeProdotto], \$_POST[marcaProdotto], \$_POST[descrizioneProdotto])		
Esempi create table SQL:		
CREATE TABLE prodotti(idProdotto int not null PRIMARY KEY auto_increment, genere varchar(30) not null, etichetta longtext not null, costo double(6,2)not null, nome varchar(50)not null, marca varchar(50)not null, descrizione longtext not null))		

4)

PUT /spesa/prodotto/				
Summary				
Modifica un prodotto				
Description				
Permette la modifica di un prodotto specificando l'id del prodotto da modificare				
Parameters				
Name	Located In	Description	Required	Schema
idProdotto	Body	ID del prodotto	Yes	Xs:int
Genere	Body	Genere del prodotto	Yes	Xs:string
Etichetta	Body	Etichetta del prodotto	Yes	Xs:string
Costo	Body	Costo del prodotto	Yes	Xs:double
Nome	Body	Nome del prodotto	Yes	Xs:string

Marca	Body	Marca del prodotto	Yes	Xs:string
Descrizione	Body	Descrizione del prodotto	Yes	Xs:string
Responses				
Code	Description	Schema		
200	Operazione eseguita correttamente	<xs:element name="return" type="xs:boolean">		
403	Non autorizzato			
406	Parametri di input non validi			
415	valore del content-type non è quello atteso			
Default	Errore non previsto			
Esempi output proposto all’utente dal consumer:				
<div>➤ “Operazione andata a buon fine”</div> <div>➤ “Errore, operazione non completata”</div>				
Esempi query SQL:				
UPDATE prodotto SET genere=\$_POST[genereProdotto], etichetta=\$_POST[etichettaProdotto], costo=\$_POST[costoProdotto], nome=\$_POST[nomeProdotto], marca=\$_POST[marcaProdotto], descrizione=\$_POST[descrizioneProdotto] WHERE idProdotto=\$_POST[idProdotto]				
Esempi create table SQL:				
CREATE TABLE prodotti(<div>idProdotto int not null PRIMARY KEY auto_increment,</div> <div>genere varchar(30) not null,</div> <div>etichetta longtext not null,</div> <div>costo double(6,2)not null,</div> <div>nome varchar(50)not null,</div> <div>marca varchar(50)not null,</div> <div>descrizione longtext not null)</div> <div>)</div>				

DELETE /spesa/prodotto/

Summary

Eliminazione di un prodotto

Description

Permette di eliminare un dato prodotto

Parameters

Name	Located In	Description	Required	Schema
idProdotto	Query	idProdotto	Yes	Xs:int

Responses

Code	Description	Schema
200	Operazione eseguita correttamente	<xs:element name="return" type="xs:boolean">
403	Non autorizzato	
406	Parametri di input non validi	
415	valore del content-type non è quello atteso	
Default	Errore non previsto	

Esempi output proposto all'utente dal consumer:

- "Operazione andata a buon fine"
- "Errore, operazione non completata"

Esempi query SQL:

```
DELETE FROM prodotto WHERE idProdotto=[idProdotto]
```

Esempi create table SQL:

```
CREATE TABLE prodotti(  
    idProdotto int not null PRIMARY KEY auto_increment,  
    genere varchar(30) not null,  
    etichetta longtext not null,  
    costo double(6,2)not null,  
    nome varchar(50)not null,  
    marca varchar(50)not null,  
    descrizione longtext not null )  
)
```